

**METHOD AND APPARATUS IN A DATA PROCESSING SYSTEM FOR  
MANAGING SITUATIONS FROM CORRELATED EVENTS**

**BACKGROUND OF THE INVENTION**

5

**1. Technical Field:**

The present invention relates generally to an improved data processing system, and in particular to a method and apparatus for managing alerts. Still more particularly, the present invention provides a method, apparatus, and computer implemented instructions for managing alerts from identified situations relating to a series of security violations.

15 **2. Description of Related Art:**

Much of the progress in computer technology in recent years has centered around inter-computer communication. In many cases, networks of small-scale computers have been used in place of mainframe computers. Sometimes, it is less expensive and more efficient for users to share data among single-user workstations and small-scale servers than it is to share computing time on a single mainframe computer.

Increases in connectivity between computers, especially through the Internet, the world's largest and most interconnected computer network, are not without costs. Increased connectivity brings with it an increased likelihood of a security breach or other malevolent activity. Put another way, the more accessible computers become, the more they will be accessed.

It is thus imperative for organizations that rely on

Docket No. AUS920010291US1

networks of computers to have effective security violation detection systems in place to prevent and remedy security compromises. In particular, where many system events that might be categorized as suspicious 5 take place, it is important to be able to sort through a large amount of event data to determine what is actually taking place. When system events are simply "dumped" to a human administrator or user, it is difficult for the human administrator to sort through and make sense of the 10 voluminous data.

After a detection of an attempt of an unauthorized access or other suspicious activity has occurred, an alert of the situation is typically displayed for an operator to see and process. Typically, the situation is 15 presented in a static manner as an alert or event with the alert remaining on the operator's console until the alert is either manually closed or a preset time period elapses causing the alert to be closed out. In a dynamic environment with a large amount of activity, this type of 20 alert handling may easily lead to an overwhelming number of alerts being displayed in which the alerts being displayed are difficult to display in terms of timeliness and relative importance.

Therefore, it would be advantageous to have an 25 improved method and apparatus for handling alerts of situations.

100-242660

**SUMMARY OF THE INVENTION**

5       The method of the present invention provides a  
method, apparatus, and computer implemented instructions  
for handling a situation in a data processing system. In  
response to detecting a situation, an aging function is  
applied to the situation. The manner in which alerts  
10      regarding the situation are presented is based on the  
aging function.

000427-000000000000

**BRIEF DESCRIPTION OF THE DRAWINGS**

The novel features believed characteristic of the  
5 invention are set forth in the appended claims. The  
invention itself, however, as well as a preferred mode of  
use, further objectives and advantages thereof, will best  
be understood by reference to the following detailed  
description of an illustrative embodiment when read in  
10 conjunction with the accompanying drawings, wherein:

**Figure 1** is a pictorial representation of a network  
of data processing systems in which the present invention  
may be implemented;

15 **Figure 2** is a block diagram of a data processing  
system that may be implemented as a server in accordance  
with a preferred embodiment of the present invention;

**Figure 3** is a block diagram illustrating a data  
processing system in which the present invention may be  
implemented;

20 **Figures 4A-4C** are diagrams illustrating a number of  
different scenarios in which attacks (or suspicious  
activity) directed at a network can occur in a preferred  
embodiment of the present invention;

25 **Figures 5A-5C** are diagrams of situation events  
presented on a graphical user interface in accordance  
with a preferred embodiment of the present invention;

**Figure 6** is a flowchart of a process used for  
processing an alert for a situation event in accordance  
with a preferred embodiment of the present invention;

30 **Figure 7** is a flowchart of a process used for  
generating an initial alert in accordance with a  
preferred embodiment of the present invention;

Docket No. AUS920010291US1

**Figure 8** is a flowchart of a process used for processing an event in accordance with a preferred embodiment of the present invention; and

**Figure 9** is a flowchart of a process used for 5 processing events for situations in accordance with a preferred embodiment of the present invention.

09042745082003

**DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT**

5       With reference now to the figures, **Figure 1** depicts a pictorial representation of a network of data processing systems in which the present invention may be implemented. Network data processing system **100** is a network of computers in which the present invention may be  
10 implemented. Network data processing system **100** contains a network **102**, which is the medium used to provide communications links between various devices and computers connected together within network data processing system **100**. Network **102** may include connections, such as wire, wireless communication links, or fiber optic cables.  
15

In the depicted example, server **104** is connected to network **102** along with storage unit **106**. In addition, clients **108**, **110**, and **112** are connected to network **102**. These clients **108**, **110**, and **112** may be, for example,  
20 personal computers or network computers. In the depicted example, server **104** provides data, such as boot files, operating system images, and applications to clients **108-112**. Clients **108**, **110**, and **112** are clients to server **104**. Network data processing system **100** may include  
25 additional servers, clients, and other devices not shown. In the depicted example, network data processing system **100** is the Internet with network **102** representing a worldwide collection of networks and gateways that use the TCP/IP suite of protocols to communicate with one another.  
30 At the heart of the Internet is a backbone of high-speed data communication lines between major nodes or host

computers, consisting of thousands of commercial, government, educational and other computer systems that route data and messages. Of course, network data processing system 100 also may be implemented as a number 5 of different types of networks, such as for example, an intranet, a local area network (LAN), or a wide area network (WAN). **Figure 1** is intended as an example, and not as an architectural limitation for the present invention.

Referring to **Figure 2**, a block diagram of a data 10 processing system that may be implemented as a server, such as server 104 in **Figure 1**, is depicted in accordance with a preferred embodiment of the present invention. Data processing system 200 may be a symmetric 15 multiprocessor (SMP) system including a plurality of processors 202 and 204 connected to system bus 206. Alternatively, a single processor system may be employed. Also connected to system bus 206 is memory controller/cache 208, which provides an interface to local memory 209. I/O bus bridge 210 is connected to system bus 20 206 and provides an interface to I/O bus 212. Memory controller/cache 208 and I/O bus bridge 210 may be integrated as depicted.

Peripheral component interconnect (PCI) bus bridge 214 connected to I/O bus 212 provides an interface to PCI 25 local bus 216. A number of modems may be connected to PCI local bus 216. Typical PCI bus implementations will support four PCI expansion slots or add-in connectors. Communications links to clients 108-112 in **Figure 1** may be provided through modem 218 and network adapter 220 30 connected to PCI local bus 216 through add-in boards.

Additional PCI bus bridges 222 and 224 provide

T00000000000000000000000000000000

Docket No. AUS920010291US1

interfaces for additional PCI local buses **226** and **228**, from which additional modems or network adapters may be supported. In this manner, data processing system **200** allows connections to multiple network computers. A 5 memory-mapped graphics adapter **230** and hard disk **232** may also be connected to I/O bus **212** as depicted, either directly or indirectly.

Those of ordinary skill in the art will appreciate that the hardware depicted in **Figure 2** may vary. For 10 example, other peripheral devices, such as optical disk drives and the like, also may be used in addition to or in place of the hardware depicted. The depicted example is not meant to imply architectural limitations with respect to the present invention.

15 The data processing system depicted in **Figure 2** may be, for example, an IBM e-Server pSeries system, a product of International Business Machines Corporation in Armonk, New York, running the Advanced Interactive Executive (AIX) operating system or LINUX operating 20 system.

With reference now to **Figure 3**, a block diagram illustrating a data processing system is depicted in which the present invention may be implemented. Data processing system **300** is an example of a client computer. Data 25 processing system **300** employs a peripheral component interconnect (PCI) local bus architecture. Although the depicted example employs a PCI bus, other bus architectures such as Accelerated Graphics Port (AGP) and Industry Standard Architecture (ISA) may be used. Processor **302** and main memory **304** are connected to PCI local bus **306** through PCI bridge **308**. PCI bridge **308** also 30 may include an integrated memory controller and cache

memory for processor **302**. Additional connections to PCI local bus **306** may be made through direct component interconnection or through add-in boards. In the depicted example, local area network (LAN) adapter **310**, SCSI host bus adapter **312**, and expansion bus interface **314** are connected to PCI local bus **306** by direct component connection. In contrast, audio adapter **316**, graphics adapter **318**, and audio/video adapter **319** are connected to PCI local bus **306** by add-in boards inserted into expansion slots. Expansion bus interface **314** provides a connection for a keyboard and mouse adapter **320**, modem **322**, and additional memory **324**. Small computer system interface (SCSI) host bus adapter **312** provides a connection for hard disk drive **326**, tape drive **328**, and CD-ROM drive **330**.

15 Typical PCI local bus implementations will support three or four PCI expansion slots or add-in connectors.

An operating system runs on processor **302** and is used to coordinate and provide control of various components within data processing system **300** in **Figure 3**. The 20 operating system may be a commercially available operating system, such as Windows 2000, which is available from Microsoft Corporation. An object oriented programming system such as Java may run in conjunction with the operating system and provide calls to the operating system 25 from Java programs or applications executing on data processing system **300**. "Java" is a trademark of Sun Microsystems, Inc. Instructions for the operating system, the object-oriented operating system, and applications or programs are located on storage devices, such as hard disk 30 drive **326**, and may be loaded into main memory **304** for execution by processor **302**.

Those of ordinary skill in the art will appreciate that the hardware in **Figure 3** may vary depending on the implementation. Other internal hardware or peripheral devices, such as flash ROM (or equivalent nonvolatile memory) or optical disk drives and the like, may be used in addition to or in place of the hardware depicted in **Figure 3**. Also, the processes of the present invention may be applied to a multiprocessor data processing system.

As another example, data processing system **300** may be a stand-alone system configured to be bootable without relying on some type of network communication interface, whether or not data processing system **300** comprises some type of network communication interface. As a further example, data processing system **300** may be a Personal Digital Assistant (PDA) device, which is configured with ROM and/or flash ROM in order to provide nonvolatile memory for storing operating system files and/or user-generated data.

The depicted example in **Figure 3** and above-described examples are not meant to imply architectural limitations. For example, data processing system **300** also may be a notebook computer or hand held computer in addition to taking the form of a PDA. Data processing system **300** also may be a kiosk or a Web appliance.

The present invention provides a method, computer program product, and apparatus for reporting possible security violations and managing these possible security violations in a network data processing system containing several individual client or server computers. The present invention employs the use of an aging or decay function combined with user configurable threshold

settings. With this mechanism, the severity of each alert may be dynamically determined and then displayed on a console or other interface.

Turning now to **Figures 4A-4C**, diagrams illustrating 5 a number of different scenarios in which attacks (or suspicious activity) directed at a network can occur in a preferred embodiment of the present invention.

Specifically, these figures illustrate a pattern of events or occurrences that may form a "situation". The 10 events making up the situation also are referred to as "situation events". In these examples, a situation is a summary or a group of events in which the number of events exceed some threshold. The threshold at which a situation occurs, may vary depending on a correlation 15 between the events and based on a severity for particular events involved. In the depicted examples, the events are those related to possible security violations or threats to components within a network data processing system.

20 In **Figure 4A**, a situation where a single source computer **400** directs a number of attacks **402** toward a single target computer **404** is illustrated. Attacks **402** may be of a single type, such as in the case of a "denial 25 of service attack," in which target computer **404** would be flooded with electronic mail or other network information from source computer **400**. Alternatively, the attacks may be of different types, such as an attempt to break into a user's account on target computer **404**, coupled with the transmission of a "Trojan horse" program via electronic 30 mail. A "Trojan horse," much like the famous Trojan horse of classical antiquity, is a computer program that appears useful, but actually contains hidden code with a

harmful effect.

Next, **Figure 4B** depicts a situation in which a single source computer **406** directs attacks **408** at several target computers **410** within the network. **Figure 4C** 5 depicts another situation in which several source computers **412** direct attacks **414** at a single target computer **416**.

One can more easily understand what is happening within the network when attacks are summarized. That is, 10 if one is told that "Computer A is attacking computers on the network," one knows immediately to do something about "Computer A." If, on the other hand, if one reads a list of all of the possibly suspicious activity in the network, one may experience information overload; one may 15 not realize that a problem exists with "Computer A." This is particularly true when "Computer A" attacks multiple computers, as in **Figure 4B**. The present invention provides a way to summarize reports of suspicious activity in such a way that an administrator 20 operating a computer system within the network can easily identify and correct security problems.

With situations, such as those illustrated in **Figures 4A-4C**, the alerts for these types of situations may be presented and dynamically changed to indicate the 25 decreasing or increasing of the severity of the situation. The aging function used by the mechanism of the present invention may be any time dependent function appropriate to a given scenario. This function may increase or decrease the severity of the alert as time 30 passes. Typical time dependent functions include functions, such as an exponential decay function with a given half-life, an exponentially increasing function,

linearly decreasing functions, and a linearly increasing function. Other possibilities include a combination of two types of behavior, such as exponentially decreasing (or increasing) function for a set time period followed by a linearly decreasing (or increasing) function.

Another useful possibility is to combine a linear or exponential function preceded or followed by time independent behavior, such as, for example, a straight line or curve followed or preceded by a flat line.

As an example, consider an application that deals with pattern or situation events, which have numerical severity level from 0 to 100 and a severity label, which may be assigned. Suppose that the severity label may take on these four values in order of increasing importance of the event: HARMLESS, MINOR, CRITICAL, SEVERE. The situation events are dynamic in that as new information arrives the numerical severity level of the events will change. Further, suppose that user-configurable threshold settings are provided. These settings allow for tuning the behavior of the system in terms of when a situation event is first created and how the situation changes dynamically from the moment the situation is created. Example threshold settings are shown in Table I below:

25

**TABLE I**

Severity Label	Threshold	Meaning
HARMLESS	0	A situation event of severity level between 0 and 20
MINOR	20	A situation event of severity level between 20 and 40
CRITICAL	40	A situation event of severity level between 40 and 80
SEVERE	80	A situation event of severity level between 80 or greater

Now when a situation is first created the severity level is set appropriately as determined by the thresholds. As new information arrives and the severity level increases or decreases, the severity label for the event changes accordingly as the various thresholds are crossed. When no new information arrives, however, the severity level and label for the event will continue to change according to a time dependent aging function.

For example, suppose that an exponential decay function is used with a half-life of two hours. When an situation reaches or is initially created with a severity level just below 80, the severity label will be CRITICAL. If no new information is arriving, the severity level will decrease over time. After two hours, the level reaches 40 and the label changes from CRITICAL to MINOR. After another two hours passes, the situation severity level reaches 20 and the severity label transitions from MINOR to HARMLESS. Variations on the scenario described above may be introduced to provide more complex and useful behavior. For example, a lower threshold setting may be used to determine when a situation event is destroyed or stopped. Additionally, when the situation event severity level drops below a value of 2, the event is deleted (or archived) and removed from the operator's display. Another variation may be to allow for a user-configurable setting to indicate when a situation event reaches a certain severity label or above, such as CRITICAL or higher. Although the numerical severity level may decrease, the severity label will remain at the highest value, which has been reached.

With reference now to **Figures 5A-5C**, diagrams of situation events presented on a graphical user interface

are depicted in accordance with a preferred embodiment of the present invention. In these examples, dynamic changes to presentations of situation events are depicted as handled through applying an aging function according  
5 to the present invention.

In **Figure 5A**, window **500** is an example of a window in a console for presenting alerts to an operator. Situation events **502**, **504**, **506**, **508**, and **510** are displayed within window **500**. In this example, situation  
10 event **502** involves a "denial of service" (DoS), while situation events **504**, **506**, **508**, and **510** indicate that Internet access to a system has been blocked. Situation event **502** is MINOR through its association with graphical indicator **512**. In **Figure 5B**, situation events **514**, **516**,  
15 **518**, and **520** are displayed in addition to situation event **502**. At this time, the severity of situation event **502** has been reduced to HARMLESS as indicated by its association with graphic indicator **522**. The severity of  
the alert for situation event **502** reduces over the  
20 passage of time through an application of an aging function to the situation event.

In **Figure 5C**, situation events **524**, **526**, **528**, **530**, and **532** are illustrated. Situation event **502** no longer appears within window **500**. At this point in time,  
25 situation event **502** no longer appears based on the amount of time that has passed when an aging function is applied to this event.

In **Figure 5A**, only a graphical indicator in the form of a bullet is illustrated for situation event **502** to  
30 more clearly describe the mechanism of the present invention. Typically, other events also may have

associated graphical indicators. Further, the indicators may take other forms, such as, for example, blinking text, changing colors in text, or changing colors for a graphical indicator.

5       Turning next to **Figure 6**, a flowchart of a process used for processing an alert for a situation event is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 6** may be implemented in a data processing system, such as  
10 data processing system 200 in **Figure 2**.

The process begins by detecting an initial alert which indicates a situation (step **600**). Next, an aging function is applied to the situation (step **602**). The particular aging function applied to the situation  
15 depends on the particular situation detected. A determination is then made as to whether an alert is present (step **604**). The determination is based on the result of applying the aging function to the situation. If an alert for a situation is present, the alert is  
20 presented on a display (step **606**) with the process returning to step **602** as described above. Otherwise, the process terminates.

For example, a Web site is subjected to floods of network traffic with valid Web server requests that tend  
25 to overwhelm the Web server(s). In a denial of service (DoS) attack, the high volume of activity may make the Web site unusable or difficult to access for normal users. The events that form this situation may be requests originating from one or more sources.

30       As the DoS attack progresses over an extended period of time, a network-based sensor tracks the network activity and generates alerts to a central server. The

alarm results in the creation of a situation on the console. Over time, the severity increases dramatically for the situation (or alarm) that is presented on the console because the problem is becoming more urgent. At 5 some point, the situation may be deemed a CRITICAL situation. In this case, an action, such as sending e-mail to an administrator or paging the administrator may be initiated based on the severity of the situation crossing the CRITICAL threshold. However, once the DoS 10 attack subsides, displaying a CRITICAL situation alert on the console indefinitely may be inappropriate. Continuing to display this alert clutters the console with events that may have already been handled or may become less interesting over time, since it reflects 15 activity that happened in the past.

To address this problem, the mechanism of the present invention may apply an exponential, time-dependent decay function with a configured half-life to the situation. For example, if the situation reached 20 a severity of 60 and then quiesced, a half-life of 2 hours would result in a severity level of 30 after 2 hours, then 15 after 4 hours and so on. At some point, the severity of the situation will reach a level below a minimum threshold and be removed from the console.

With reference now to **Figure 7**, a flowchart of a process used for generating an initial alert is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 7** may be implemented in a data processing system, such as data 30 processing system **200** in **Figure 2**.

The process begins by collecting data on events (step **700**). Next, summaries are generated from the

TOP SECRET//SI//FOUO

Docket No. AUS920010291US1

events (step 702). An unprocessed summary is selected from the summaries for processing (step 704). A determination is then made as to whether the summary exceeds a threshold (step 706). If the summary exceeds 5 the threshold, an initial alert is generated (step 708). Then a determination is made as to whether more unprocessed summaries are present (step 710). If more unprocessed summaries exist, the process returns to step 704 as described above. Otherwise, the process 10 terminates. Turning back to step 706, if the summary does not exceed the threshold, the process proceeds to step 710 as described above. A more detailed description of grouping events into summaries and generating alerts is found in *Presentation of Correlated Events as 15 Situation Classes*, attorney docket no. AUS920010242US1, application no. \_\_\_\_\_, filed even date hereof and incorporated herein by reference.

Turning next to **Figure 8**, a flowchart of a process used for processing an event is depicted in accordance 20 with a preferred embodiment of the present invention. The process illustrated in **Figure 8** may be implemented in a data processing system, such as data processing system 200 in **Figure 2**.

The process begins by detecting an event (step 800). 25 A determination is made as to whether the event is a first event for a situation (step 802). If the event is not a first event for the situation, the severity of the situation is increased (step 804) with the process terminating thereafter. If the event is a first event 30 for the situation, an alert is created on a console (step 806) with the process terminating thereafter. This step

is used to identify whether this particular event is the event that triggers the beginning of a situation. This event may follow other events, which may be part of the situation but insufficient to trigger the situation.

5 This example applies a linear aging function to events for a situation as part of the process for increasing the severity of the alert for the situation. Any aging function may be used depending on the particular situation for which events are being processed.

10 With reference now to **Figure 9**, a flowchart of a process used for processing events for situations is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 9** may be implemented in a data processing system, such as  
15 data processing system 200 in **Figure 2**. The process illustrated in this figure may be used for evaluating all situations on a timed basis.

The process begins by detecting an expiration of a timer (step 900). Next, alerts are identified for  
20 situations (step 902). Then, an unprocessed alert for a situation is selected (step 904). Next, a determination is made as to whether the situation is subject to an increasing time function (step 906). If the situation is subject to an increasing time function, the alert is  
25 adjusted using a linear function (step 908). A determination is then made as to whether more unprocessed alerts for situations are present (step 910). If additional unprocessed alerts are present, the process returns to step 904 as described above. Otherwise the  
30 process terminates. Turning back to step 906, if the situation associated with the alert is not subject to an increasing time function, the alert is adjusted using a

half-life function (step 912) with the process proceeding to step 910 as described above.

This depicted example illustrates a selection from two types of aging functions for purposes of 5 illustrations. The mechanism of the present invention may select from other aging functions other than those shown in **Figure 9**. For example, an exponentially increasing function may be used. As time passes, the severity increases at an exponential rate. This is 10 indicative of an alert that demands immediate attention. Another function is a stepping function in which the severity decreases by a certain percentage for each unit of time. For example, the severity might decrease 25% after each 4 hour time period.

15 Thus, the present invention provides an improved method, apparatus, and computer implemented instructions for dynamically managing alerts for situations. The mechanism of the present invention applies an aging function identified by the alerts and adjusts the 20 severity of the alert based on the results. The severity may increase or decrease depending on the passage of time and what events are detected for different situations. The mechanism of the present invention allows for a 25 reduction in the number of alerts displayed to a user by removing alerts for situations that fall below some minimal threshold. The mechanism of the present invention may be applied to other types of situations other than those involving a denial of service.

For example, the mechanism of the present invention 30 also may be applied to suspicious Web server requests. In this example, a relatively small number of requests are sent to a Web server by an individual. The requests

are highly suspicious because these requests are designed to attempt to access information that should not be accessible. The fact that the suspicious requests are made is serious and results in a situation being created

5 on the console. If the request is actually successful, for example, the sensor determines the user actually was able to access the information, then the severity of the situation is increased again. The question then arises as to how to handle the situation on the console if it

10 does not receive attention. The recommendation in this case is to continue to increase the severity based on a linearly increasing function over time. The increase in severity is made because the hacker has exploited a weakness in the security apparatus of the Web site, and

15 the problem becomes more severe the longer the weakness remains in place. As the severity increases over time, increasingly "vocal" mechanisms are invoked to bring the matter to the attention of an administrator. For example, these mechanisms may include e-mail, pager,

20 and/or flashing red lights.

It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the distribution. Examples of computer readable media include recordable-type media, such as a floppy disk, a hard disk drive, a RAM, CD-ROMs, DVD-ROMs, and

transmission-type media, such as digital and analog communications links, wired or wireless communications links using transmission forms, such as, for example, radio frequency and light wave transmissions. The  
5 computer readable media may take the form of coded formats that are decoded for actual use in a particular data processing system.

The description of the present invention has been presented for purposes of illustration and description,  
10 and is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. For example, the events described in the examples are those related to possible security  
15 violations or threats to components within a network data processing system. Of course, the mechanism of the present invention may be applied to other types of events other than security related events. The mechanism of the present invention may be applied to resource events. For  
20 example, an event is received that a database server is running low on disk space. Another type is for 911 calls, such as those for emergencies requiring police or fire services. 911 calls are posted on a console and based on information collected on the call, the algorithm  
25 for managing severity on the console might be adjusted.

The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the invention for various  
30 embodiments with various modifications as are suited to the particular use contemplated.